

# SpikingNeRF: Making Bio-inspired Neural Networks See through the Real World (Appendix)

Xingting Yao<sup>1,2\*</sup>, Qinghao Hu<sup>1\*</sup>, Fei Zhou<sup>3</sup>, Tielong Liu<sup>1,2</sup>,  
Zitao Mo<sup>1</sup>, Zeyu Zhu<sup>1,2</sup>, Zhengyang Zhuge<sup>1</sup>, Jian Cheng<sup>1†</sup>

<sup>1</sup>Institute of Automation, Chinese Academy of Sciences

<sup>2</sup>School of Future Technology, University of Chinese Academy of Sciences

<sup>3</sup>China Electric Power Research Institute Co., Ltd

{yaoxingting2020, huqinghao2014, jian.cheng}@ia.ac.cn,

## Technical appendix guideline

Following is the Technical appendix to the main text, we introduce how this technical appendix is organized in this section. In **Additional implementation details**, we introduce more details of datasets and more details of the SpikingNeRF implementation on the DVGO (Sun, Sun, and Chen 2022) and the TensoRF (Chen et al. 2022) framework, where the principles for experimental fairness are strictly adhered to. In **Energy consumption estimation**, we illustrate how to estimate the theoretical energy consumption in our experiments. In **Evaluation details on SpikeSim**, we describe evaluation details on the SpikeSim benchmark, and we further showcase and discuss the evaluation on GPU and SATA (Yin et al. 2022). In **Additional experiment results**, we reveal more results of our main experiments and ablation study, including the fairness experiments, the ineffective Poisson-encoding results and extensive statistics for the ablation study. In **Extensive experiments**, we demonstrate the effectiveness of SpikingNeRF on different 3D tasks, and apply SpikingNeRF to the TensoRF framework for extensive verification. In **Reproducibility**, we introduce how to reproduce the main experiments with the provided codes.

## Additional implementation details

All our experiments are conducted on A100 GPUs, and the reconstruction process (both training and inference) of each scene is assigned to single GPU. In **Reproduction results for fairness**, we further demonstrate the fairness of our implementation by experimentation. *All experiments use the same default random seed and are not randomness-dependent, so there is no need to do several runs. Following prior arts of NeRF, we run each experiment for one time.*

## Dataset introduction

Here, we give a more sufficient introduction of the datasets we use in the paper. We conduct experiments mainly on the four inward-facing datasets, including Synthetic-NeRF (Mildenhall et al. 2021) that contains eight objects synthesized from realistic images, Synthetic-NSVF (Liu et al. 2020) that contains eight objects synthesized by NSVF,

BlendedMVS (Yao et al. 2020) with authentic ambient lighting by blending real images, and Tanks&Temples (Knapitsch et al. 2017) which is a real world dataset.

## Implementation on DVGO

As described in the main text, we produce both SpikingNeRF-D and DVGO results with the DVGO official codes. In the inward-facing bounded task, including Synthetic-NeRF, Synthetic-NSVF, BlendedMVS, and Tanks&Temples, we keep all the hyper-parameters as same as the original configuration (Sun, Sun, and Chen 2022) except for the training iteration in the fine stage raised from 20000 to 40000 because SNNs usually encounter the under-fitting issue that requires more training iterations to resolve (Fang et al. 2021b,a). The comparison between SpikingNeRF-D and DVGO is fair since the training iteration of our reproduced DVGO is also set to 40000. In the inward-facing unbounded and the forward-facing task, we also keep all hyper-parameters, the same with the original configurations for both SpikingNeRF-D and our reproduced DVGO. Notably, the grid resolution setting has different options in the DVGO paper, but this paper chooses only one of them, which is 160<sup>3</sup>.

## Implementation on TensoRF

We also plug in our SpikingNeRF method to the official TensoRF codes (Chen et al. 2022). For both SpikingNeRF-T and our reproduced TensorRF, we discard the feature embedding to alleviate the encoding layer's computation burden. Except that, all hyper-parameters are the same as the official configurations. Finally, In **Reproducibility**, we show how to use our provided codes to reproduce our main experiments.

## Energy consumption estimation

We follow the prior arts (Zhou et al. 2022; Yao et al. 2023; Kundu et al. 2021; Kundu, Pedram, and Beerel 2021; Horowitz 2014) to provide the theoretical energy consumption estimation in 45nm technology (Horowitz 2014), and report the energy consumption of rendering a novel view on average. The specific method is as follows:

First, the energy cost of the spike-based operations in

\*Equal contribution.

†Corresponding author.

SNNs are defined as:

$$SOPs = Spike\_num \times Flops, \quad (1)$$

$$Energy_{SOPs} = 0.9pJ \times SOPs, \quad (2)$$

where  $Spike\_num$  denotes the spike number of the input spike train,  $Flops$  is the flop-point operations of the following computation that one spike will trigger, and  $SOPs$  is the number of spike-based operations.

Similarly, the energy consumption of the flop-point operations in ANNs is:

$$Energy_{FLOPs} = 4.6pJ \times FLOPs, \quad (3)$$

where  $FLOPs$  is the number of the flop-point operations.

For a specific NeRF-based work, the total energy consumption of a novel view  $Energy_{tot}$  is obtained through:

$$Energy_{tot} = Pts\_num \times (Energy_{SOPs} + Energy_{FLOPs}), \quad (4)$$

where  $Pts\_num$  is the total number of the sampled points.

For the mask-free methods (Mildenhall et al. 2021; Barron et al. 2021; Deng, Barron, and Srinivasan 2020), we can easily obtain the  $Pts\_num$  for the number of sampling is fixed. For the methods with the masking operation (Wu et al. 2022; Sun, Sun, and Chen 2022; Chen et al. 2022) and SpikingNeRF, we count the  $Pts\_num$  during the runtime. Specially for NSVF (Liu et al. 2020) that uses the masking operation but costs days to train on single scene and does not provide pretrained models, we can hardly afford the computation overhead of training and choose to estimate the  $Pts\_num$  through:

$$Pts\_num = \rho \cdot Pts\_Num, \quad (5)$$

where  $Pts\_Num$  is the samples' number before the masking operation, which is fixed, and  $\rho$  is the valid sample ratio obtained through our DVGO practice and the statistics from (Hu et al. 2022). For Synthetic-NeRF, Synthetic-NSVF, BlendedMVS, and Tanks&Temples, we set  $\rho$  to 4.04%, 2.18%, 5.39%, and 7.70%, respectively. Such an estimation on  $Pts\_num$  in Eq. 5 will incur numeric errors between the runtime-counted (real) and the estimated energy cost on NSVF (Liu et al. 2020).

Given all the aforementioned aspects, we exclusively test this estimation method on Synthetic-NeRF and report “Runtime-counted(mJ) v.s. Estimated(mJ)”, which turns out to be “45832 v.s. 16427”. Despite the error rate being 64.16%, both the Runtime-counted and Estimated energy costs remain at the same numeric magnitude. While, the gap between SpikingNeRF-D and NSVF in energy consumption spans orders of magnitudes as illustrated in the main text. Therefore, the advantages of SpikingNeRF over NSVF in energy consumption should be considered definitive. Note that, we only report the estimated energy cost of NSVF, and the energy costs of other mask-applied works are all runtime-counted.

With the above methods, we can quantitatively compare SpikingNeRF with its ANN counterparts in energy consumption to show the energy merits SpikingNeRF brings.

## Evaluation details on SpikeSim

The energy estimation described in **Energy consumption estimation** is commonly used to compare SNNs and their ANN counterparts on the same hardware technology. However, our proposed TCP can further improve the energy efficiency and the inference speed of SpikingNeRF on SNN-tailored hardware accelerators.

To demonstrate the advantages of TCP over TP, we benchmark SpikingNeRF-D on SpikeSim with the SpikeFlow architecture (Moitra et al. 2023). SpikeSim is an neuromorphic accelerator based on the in-memory-processing technology. It stores the parameters as the conductance of the crossbars in each processing unit, each crossbar is responsible for a metric-vector multiplication at a time, and each processing unit will handle several metric-vector multiplications in parallel. Since SpikingNeRF-D’s sMLP is very small, we set the crossbar size from the original  $64 \times 64$  to  $32 \times 32$ , thus the mapping from the sMLP to the SpikeFlow architecture becoming feasible. Besides, the original SpikeFlow architecture is designed for image classification tasks and does not consider dataflow’s parallelism because the original model size is big. So, in our practice, we consider the parallelism of dataflow being set to 4, which means four sMLPs are stored on chip and four rays of radiance fields can be queried by the sMLPs simultaneously. Thus, the evaluation of both TCP and TP is feasible on SpikeSim.

## More evaluations and discussions on GPUs and SATA for the demonstration of the effectiveness of TCP

To further demonstrate the effectiveness of our proposed TCP, we further discuss how TCP can be so beneficial to hardware. **A)** For any hardware accelerator and even for GPUs (Tab. 3), a regular and condensed data structure commonly brings far more benefits to efficiency even if condensing has extra overhead. Such benefits also apply to neuromorphic hardware, as proposed and proved in (Lee, Zhang, and Li 2022) (published at a tier-1 hardware conference). **B)** Not all neuromorphic hardware is event-driven, i.e., sparsity-aware, taking SpikeSim for example. This hardware is not able to skip zero data and has to strictly follow the predefined data flow. That’s why TCP producing a more effective data flow could benefit greatly on SpikeSim (65.78 vs 559.45). Such benefit is quantitatively correlated to the Sample Execution Ratio (SER), which refers to the ratio of samples involved in NeRF rendering. As listed in Tab. 3, 100% samples are processed in TP where zero data are interspersed within while TCP only requires 11.78% where zero data are condensed and discarded. **C)** Even for event-driven hardware (e.g., SATA) that can skip zero data, “no event” (zero data) still costs energy to maintain the dynamics update and sparsity-aware modules. In this hardware, TCP, i.e. low SER, would not benefit so greatly but still help a lot as listed in Tab. 3 (8.70 vs 25.55).

Additionally, in the SATA evaluation, we do not change the default configuration of the official implementation. We simply mapping the SpikingNeRF-D to SATA and report the average energy consumption of the ray rendering process.

Table 1: Comparisons between our reproductions and the official results.

Dataset Metric	Synthetic-NeRF		Synthetic-NSVF		BlendedMVS		Tanks&Temples	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Official results collected from their papers.								
DVGO	31.95	0.957	35.08	0.975	28.02	0.922	28.41	0.911
TensoRF	33.14	0.963	36.52	0.982	-	-	28.56	0.920
Our reproduced results.								
DVGO*	31.98	0.957	35.12	0.976	28.15	0.922	28.42	0.912
TensoRF*	31.14	0.963	36.74	0.982	-	-	28.50	0.920

\* denotes our reproduced results by the official codes.

Table 2: Comparisons between the TCP-based and the TP-based SpikingNeRF-D with more metrics.

Dataset Method	Synthetic-NeRF			Synthetic-NSVF			BlendedMVS			Tanks&Temples		
	TCP	TP	DVGO*	TCP	TP	DVGO*	TCP	TP	DVGO*	TCP	TP	DVGO*
PSNR↑	31.34	31.34	31.98	34.33	34.33	35.12	27.80	27.80	28.15	28.09	28.01	28.42
SSIM↑	0.949	0.949	0.957	0.970	0.970	0.976	0.912	0.912	0.922	0.896	0.892	0.912
LPIP <sub>Vgg</sub> ↓	0.068	0.068	0.053	0.039	0.039	0.032	0.103	0.103	0.099	0.172	0.174	0.153
LPIP <sub>Alex</sub> ↓	0.039	0.039	0.034	0.021	0.021	0.018	0.065	0.065	0.070	0.139	0.142	0.143
Flops (G)	19.34	19.47	81.46	9.89	10.01	40.84	16.79	16.95	69.71	102.92	85.44	466.93
Sops (G)	24.26	24.49	-	12.43	12.80	-	21.24	21.57	-	119.57	100.52	-
Energy (mJ)	110.80	111.59	374.72	56.69	57.57	187.85	96.37	97.38	320.66	581.04	483.47	2147.86

\* denotes our reproduced results by the official codes.

Table 3: Comparisons of SpikingNeRF (sNeRF) on different platforms on Synthetic-NeRF.

GPU	PSNR	Train. Mins	Infer. Secs
ANN Counterpart	31.98	6.02	0.21
sNeRF-D TP	31.34	61.28	1.22
sNeRF-D TCP	31.34	<b>16.74</b>	<b>0.44</b>
Neuromorphic Hardware	SER	Energy+	Energy-
sNeRF-D TP	100%	559.45	25.55
sNeRF-D TCP	<b>11.78%</b>	<b>65.78</b>	<b>8.70</b>

+ and - denote the energy produced by SpikeSim and SATA (Yin et al. 2022), respectively.

## Additional experiment results

### Reproduction results for fairness

Tab. 1 lists the results of our reproduced DVGO and TensoRF along with the official results to confirm the fairness and correctness of our experiments concerning the ANN counterparts. All experiments use the same default random seed, so there is no need to do several runs, following prior arts of NeRF.

### More results under different metrics

As listed in Tab. 2, we report more experimental results of SpikingNeRF-D and DVGO under different metrics. Most rendering energy of SpikingNeRF-D is consumed in the encoding layer where float-operation is performed. And, the synthesis quality of SpikingNeRF-D falls behind the ANN counterpart by a small margin under different quality metrics. Notably, the proposed TCP and TP show the same-level synthesis quality and energy consumption. That is why a more accurate estimation on SpikeSim is needed to demonstrate the advantage of TCP.



Figure 1: Qualitative comparisons on the different challenging parts. **Top:** On *Character* from BlendedMVS, where the color changes densely and intensely. **Middle:** On *Ignatius* from Tanks&Temples, where the textures are distinct and dense. **Bottom:** On *Truck* from Tanks&Temples, where detail information are explicitly displayed. Full visualizations are deferred to the appendix.

### The ineffective Poisson-encoding

To avoid the over-length of the main text, we move the results of the ineffective Poisson-encoding to this section. As listed in Tab. 4, the Poisson-encoding scheme shows at most 24.83 PSNR among all the four datasets, and none of these results shows valid 3D rendering quality. Even with the increasing time steps, the synthesis performance will not climb up as the direct-encoding and the proposed TRA do.

We also visualize two classic rendering outcomes in Fig. 3. Obviously, Poisson-encoding tends to produce novel scenes with a single major color. Green for *Chair* and red for *Character*. This further proves the incapability of Poisson-

encoding in rendering real RGB scenes.

## Qualitative results

As shown in Fig. 4, we provide full results of synthesized novel views with the original resolution. And in Fig. 1, we provide detailed visualizations of synthesized novel views. Just as concluded in the main text, SpikingNeRF-D shares the same-level synthesis quality as DVGO.

As mentioned above, the ineffective Poisson-encoding is also discussed and visualized in **The ineffective Poisson-encoding** and Fig. 3.

## Comparisons with direct-encoding on specific scenes

In the main text, we only show the averaged PSNR, SSIM, and energy of our SpikingNeRF. To demonstrate the overall merits of our proposed TRA and reveal more experimental results, we showcase every PSNR, SSIM, and energy of SpikingNeRF-D with TRA and direct-encoding on each scene of Synthetic-NeRF as the extensive results for the ablation study of **”Comparisons with the conventional data encodings”**.

In Tab. 8 and Tab. 9, we compare time-ray alignment (TRA) with direct-encoding (DE) with different time steps on each scene of Synthetic-NeRF. In Tab. 10 and Tab. 11, we compare time-ray alignment (TRA) with direct-encoding (DE) with different density levels on each scene of Synthetic-NeRF. These four tables extend the statistics of the original two tables of the main text and still prove our proposed TRA can consistently outperform the conventional direct-encoding scheme.

## Effects of temporal flip

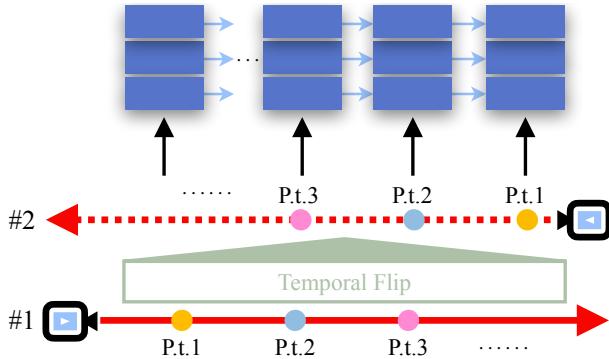


Figure 2: Temporal flip. The direction of the temporal dimension is consistent with the ray #1 but opposite to the ray #2. “P.t.” is the abbreviation of “point”.

Fig. 2 vividly illustrates the temporal-flip operations. As described in the main text, using the temporal flip means the sequential processing direction of SNNs is opposite to the radiance ray. In Tab. 5, we show the results of the TP-based SpikingNeRF-D with and without temporal flip. Although, temporal flip slightly improves the TP-based SpikingNeRF-D on synthesis quality on Tanks&Temples. In most cases,

temporal flip incurs defeats on synthesis quality and energy cost. Therefore, as concluded in the main text, the consistent alignment direction is better for SpikingNeRF.

## Extensive experiments

### Extension to other 3D tasks

As shown in Tab. 6, we also conduct experiments on SpikingNeRF-D on the inward-facing unbounded dataset Mip-NeRF360 (Barron et al. 2022) and the forward-facing dataset LLFF (Mildenhall et al. 2019) to demonstrate the effectiveness of SpikingNeRF on different kinds of 3D reconstruction. In terms of synthesis quality, SpikingNeRF-D works well with a small margin to the ANN counterpart as verified before. As for the energy merits, SpikingNeRF-D outperforms DVGO by 60.66% and 32.78% in energy reduction on Mip-NeRF360 and LLFF, respectively.

### Extension to TensoRF

In Tab. 7, we list detailed results of SpikingNeRF-T and our reproduced TensoRF. Similar to the case of SpikingNeRF-D and DVGO, SpikingNeRF-T achieves significant energy reduction with small synthesis performance drop. In addition to the main text, SpikingNeRF-T obtains 64.90% energy reduction with 1.07 PSNR drop on LLFF.

## Reproducibility

For convenience to read and use, we write down every single command used to build the experimental environment, launch the experiments, and store the experimental results in the MARKDOWN style in the README.md file. And, we have also added very detailed instructions and descriptions to each command. One should easily reproduce our main experiment results with our provided codes and README.md.

If any bug occurs due to a different environment (e.g. a different CUDA version), one may refer to the official DVGO web to seek help. Because we build most of our codes on DVGO, as claimed in the main text.

Note that, we are not able to attach our pretrained model checkpoint due to the file size limitation. We are not able to attach any training logs as well because one of the authors’ name is printed in these logs.

Table 4: Results of Poisson-encoding on different datasets with different time steps.

Dataset Metric	Synthetic-NeRF		Synthetic-NSVF		BlendedMVS		TanksTemples	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Poisson-encoding T=1	22.03	0.854	24.83	0.893	20.74	0.759	21.53	0.810
Poisson-encoding T=2	21.98	0.855	24.83	0.893	20.74	0.759	21.57	0.814
Poisson-encoding T=4	21.90	0.856	24.83	0.893	20.74	0.759	21.60	0.818

T denotes the time step.

Table 5: Comparisons with temporal flip on TP-based SpikingNeRF-D.

Dataset SpikingNeRF-D	Synthetic-NeRF		Synthetic-NSVF		BlendedMVS		Tanks&Temples	
	w/o TF	w/ TF	w/o TF	w/ TF	w/o TF	w/ TF	w/o TF	w/ TF
PSNR↑	<b>31.34</b>	31.24	<b>34.34</b>	34.06	<b>27.80</b>	27.79	28.01	<b>28.06</b>
SSIM↑	<b>0.949</b>	0.947	<b>0.970</b>	0.967	<b>0.912</b>	0.909	0.892	<b>0.894</b>
Energy (mJ)	<b>111.59</b>	117.43	57.57	<b>56.75</b>	<b>97.38</b>	105.51	<b>483.48</b>	617.57

Table 6: Comparisons on other 3D tasks.

Dataset Method	Mip-NeRF360		LLFF	
	SpikingNeRF-D	DVGO*	SpikingNeRF-D	DVGO*
PSNR↑	24.66	25.41	25.13	26.04
SSIM↑	0.646	0.694	0.792	0.834
LPIP <sub>Vgg</sub> ↓	0.461	0.430	0.261	0.210
LPIP <sub>Alex</sub> ↓	0.364	0.372	0.158	0.133
Energy (mJ)	831.43	2113.46	235.12	349.80

\* denotes our reproduced results by the official codes.

Table 7: Extension to the TensoRF framework.

Dataset Method	Synthetic-NeRF		Synthetic-NSVF		Tanks&Temples		LLFF	
	SpikingNeRF-T	TensoRF*	SpikingNeRF-T	TensoRF*	SpikingNeRF-T	TensoRF*	SpikingNeRF-T	TensoRF*
PSNR↑	32.45	33.14	35.76	36.74	28.09	28.50	25.63	26.69
SSIM↑	0.956	0.963	0.978	0.982	0.904	0.920	0.808	0.835
LPIP <sub>Vgg</sub> ↓	0.031	0.028	0.014	0.011	0.124	0.124	0.134	0.116
LPIP <sub>Alex</sub> ↓	0.058	0.047	0.031	0.025	0.163	0.141	0.227	0.203
Flops (G)	42.10	139.39	25.95	101.11	207.01	606.53	548.67	2033.68
Sops (G)	52.42	-	34.03	-	237.39	-	844.20	-
Energy (mJ)	240.81	641.17	149.98	465.09	1165.90	2790.03	3283.70	9354.94

\* denotes our reproduced results by the official codes.



Figure 3: Qualitative results of SpikingNeRF-D with Poisson-encoding and different time steps on *Chair of Synthetic-NeRF*(top) and *Character of BlendedMVS*(bottom).

Table 8: Comparisons between TRA and Direct-encoding (DE) with different time steps on the first four scenes of Synthetic-NeRF.

Scene Metric	Drums			Ficus			Hotdog			Ship		
	PSNR↑	SSIM↑	Energy↓									
TRA D=1	<b>25.21</b>	<b>0.922</b>	66.99	32.04	0.973	<b>47.24</b>	<b>36.01</b>	<b>0.973</b>	<b>148.49</b>	<b>28.63</b>	<b>0.867</b>	<b>283.45</b>
TRA D=2	<b>25.36</b>	<b>0.925</b>	<b>120.21</b>	<b>32.29</b>	0.975	<b>84.80</b>	<b>36.19</b>	<b>0.975</b>	<b>234.84</b>	28.81	<b>0.871</b>	<b>468.67</b>
TRA D=4	<b>25.39</b>	<b>0.926</b>	<b>218.28</b>	<b>32.39</b>	<b>0.976</b>	<b>151.11</b>	<b>36.23</b>	0.975	<b>382.15</b>	<b>28.86</b>	<b>0.873</b>	<b>773.86</b>
DE T=1	25.19	0.921	<b>66.54</b>	<b>32.07</b>	0.973	47.94	35.68	0.970	157.66	28.57	0.865	291.80
DE T=2	25.28	0.923	128.60	32.26	0.975	93.86	36.03	0.974	278.16	28.81	0.869	575.61
DE T=4	25.31	0.924	272.50	32.24	0.974	192.86	36.21	0.975	561.47	28.77	0.870	1190.38

Table 9: Comparisons between TRA and Direct-encoding (DE) with different time steps on the last four scenes of Synthetic-NeRF.

Scene Metric	Lego			Materials			Mic			Chair		
	PSNR↑	SSIM↑	Energy↓	PSNR↑	SSIM↑	Energy↓	PSNR↑	SSIM↑	Energy↓	PSNR↑	SSIM↑	Energy↓
TRA D=1	<b>33.82</b>	<b>0.968</b>	<b>93.20</b>	<b>29.15</b>	<b>0.942</b>	<b>166.88</b>	32.47	0.977	<b>26.48</b>	<b>33.36</b>	<b>0.969</b>	53.67
TRA D=2	<b>34.16</b>	<b>0.971</b>	<b>150.45</b>	29.27	0.944	<b>305.32</b>	<b>32.85</b>	0.979	<b>43.04</b>	<b>33.69</b>	0.972	<b>81.21</b>
TRA D=4	<b>34.22</b>	<b>0.971</b>	<b>253.52</b>	29.21	0.942	<b>481.77</b>	<b>32.98</b>	0.980	<b>74.48</b>	<b>33.81</b>	<b>0.973</b>	<b>135.56</b>
DE T=1	33.53	0.965	96.35	29.05	0.938	167.06	<b>32.49</b>	0.977	26.64	33.15	0.968	<b>50.28</b>
DE T=2	34.06	0.970	172.93	<b>29.28</b>	0.944	314.55	32.76	0.979	48.69	33.60	0.972	85.13
DE T=4	34.07	0.970	351.84	<b>29.30</b>	<b>0.944</b>	645.10	32.84	0.980	100.08	33.67	0.972	176.29

Table 10: Comparisons between TRA and Direct-encoding (DE) with density levels on the first four scenes of Synthetic-NeRF.

Scene Metric	Drums			Ficus			Hotdog			Ship		
	PSNR↑	SSIM↑	Energy↓									
TRA D=1	<b>25.21</b>	<b>0.922</b>	66.99	32.04	0.973	<b>47.24</b>	<b>36.01</b>	<b>0.973</b>	<b>148.49</b>	<b>28.63</b>	<b>0.867</b>	<b>283.45</b>
TRA D=2	<b>25.36</b>	<b>0.925</b>	<b>120.21</b>	<b>32.29</b>	<b>0.975</b>	84.80	<b>36.19</b>	<b>0.975</b>	<b>234.84</b>	28.81	<b>0.871</b>	<b>468.67</b>
TRA D=4	<b>25.39</b>	<b>0.926</b>	<b>218.28</b>	<b>32.39</b>	<b>0.976</b>	<b>151.11</b>	<b>36.23</b>	<b>0.975</b>	<b>382.15</b>	<b>28.86</b>	<b>0.873</b>	<b>773.86</b>
DE D=1	25.19	0.921	<b>66.54</b>	<b>32.07</b>	0.973	47.94	35.68	0.970	157.66	28.57	0.865	291.80
DE D=2	25.27	0.923	<b>119.08</b>	32.21	0.974	<b>83.10</b>	35.78	0.975	259.51	28.79	0.869	491.15
DE D=4	25.31	0.923	<b>211.05</b>	32.22	0.974	<b>149.26</b>	35.88	0.972	451.66	28.80	0.870	840.95

Table 11: Comparisons between TRA and Direct-encoding (DE) with density levels on the last four scenes of Synthetic-NeRF.

Scene Metric	Lego			Materials			Mic			Chair		
	PSNR↑	SSIM↑	Energy↓	PSNR↑	SSIM↑	Energy↓	PSNR↑	SSIM↑	Energy↓	PSNR↑	SSIM↑	Energy↓
TRA D=1	<b>33.82</b>	<b>0.968</b>	<b>93.20</b>	<b>29.15</b>	<b>0.942</b>	<b>166.88</b>	32.47	0.977	<b>26.48</b>	<b>33.36</b>	<b>0.969</b>	53.67
TRA D=2	<b>34.16</b>	<b>0.971</b>	<b>150.45</b>	<b>29.27</b>	<b>0.944</b>	<b>305.32</b>	<b>32.85</b>	0.979	<b>43.04</b>	<b>33.69</b>	0.972	81.21
TRA D=4	<b>34.22</b>	<b>0.971</b>	<b>253.52</b>	<b>29.21</b>	<b>0.942</b>	<b>481.77</b>	<b>32.98</b>	<b>0.980</b>	<b>74.48</b>	<b>33.81</b>	<b>0.973</b>	<b>135.56</b>
DE D=1	33.53	0.965	96.35	29.05	0.938	167.06	<b>32.49</b>	0.977	26.64	33.15	0.968	<b>50.28</b>
DE D=2	33.83	0.968	156.82	29.04	0.939	306.97	32.74	0.979	46.13	33.52	0.971	<b>79.74</b>
DE D=4	33.93	0.969	272.69	29.08	0.939	557.81	32.78	0.979	81.42	33.64	0.972	138.12



Figure 4: Qualitative results of SpikingNeRF-D and the ANN counterpart. Each row displays the two scenes from the same dataset. BlendedMVS, Synthetic-NeRF, and BlendedMVS are placed from top to bottom by order.

## References

- Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5855–5864.
- Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.
- Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 333–350. Springer.
- Deng, B.; Barron, J. T.; and Srinivasan, P. P. 2020. JaxNeRF: an efficient JAX implementation of NeRF. URL <http://github.com/google-research/google-research/tree/master/jaxnerf>.
- Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021a. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34.
- Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; and Tian, Y. 2021b. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2661–2671.
- Horowitz, M. 2014. 1.1 computing’s energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, 10–14. IEEE.
- Hu, T.; Liu, S.; Chen, Y.; Shen, T.; and Jia, J. 2022. Efficientnerf efficient neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12902–12911.
- Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4): 1–13.
- Kundu, S.; Datta, G.; Pedram, M.; and Beerel, P. A. 2021. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3953–3962.
- Kundu, S.; Pedram, M.; and Beerel, P. A. 2021. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5209–5218.
- Lee, J.-J.; Zhang, W.; and Li, P. 2022. Parallel time batching: Systolic-array acceleration of sparse spiking neural computation. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 317–330. IEEE.
- Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33: 15651–15663.
- Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4): 1–14.
- Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.
- Moitra, A.; Bhattacharjee, A.; Kuang, R.; Krishnan, G.; Cao, Y.; and Panda, P. 2023. SpikeSim: An end-to-end Compute-in-Memory Hardware Evaluation Tool for Benchmarking Spiking Neural Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1–1.
- Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5459–5469.
- Wu, L.; Lee, J. Y.; Bhattacharjee, A.; Wang, Y.-X.; and Forsyth, D. 2022. Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16200–16209.
- Yao, M.; Zhao, G.; Zhang, H.; Hu, Y.; Deng, L.; Tian, Y.; Xu, B.; and Li, G. 2023. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*.
- Yao, Y.; Luo, Z.; Li, S.; Zhang, J.; Ren, Y.; Zhou, L.; Fang, T.; and Quan, L. 2020. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1790–1799.
- Yin, R.; Moitra, A.; Bhattacharjee, A.; Kim, Y.; and Panda, P. 2022. Sata: Sparsity-aware training accelerator for spiking neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(6): 1926–1938.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; and Yuan, L. 2022. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*.